

Conan: a platform for complex network analysis

Ricardo Honorato-Zimmer * [†] Bryan Reynaert *

Ignacio Vergara * Tomas Perez-Acle * ^{†‡}

December 2, 2010

Abstract

Conan is a C++ library created for the accurate and efficient modelling, inference and analysis of complex networks. It implements the generation and modification of graphs according to several published models, as well as the unexpensive computation of global and local network properties. Other features include network inference and community detection. Furthermore, Conan provides a Python interface to facilitate the use of the library and its integration in currently existing applications.

Conan is available at <http://github.com/rhz/conan/>.

1 Introduction

The purpose of complex systems science is to understand systems in which a description of their components is insufficient to explain the system's behaviour. These kind of systems are found at the centre of several fields of science, such as physics, biology, and sociology [1, 16]. Networks are a common representation of those systems, that hold relevant information to study them. Thus, progress can be achieved in a broad variety of problems relating complex systems throughout the characterisation, simulation and prediction of the networks structure and dynamics. Research on this topic is an ongoing effort that has already provided promising results [5, 7, 8]. However, in spite of the continuous publication of many network-related algorithms and tools, a unified and modular platform allowing the implementation and use of these theoretical advances is lacking.

There are current applications capable of performing some of the aforementioned tasks, however, they have two main disadvantages. First, they are user-oriented, difficulting the integration into high-throughput data analysis workflows. Second, they are inflexible solutions, not suitable to tackle general problems. In order to address these shortcomings, we developed Conan, an

*Computational Biology Lab, Center for Mathematical Modeling, Faculty of Physical and Mathematical Sciences, Universidad de Chile. Av. Blanco Encalada 2120, Santiago, Chile.

[†]Fundación Ciencia para la Vida. Avda. Zañartu 1482, Ñuñoa, Santiago, Chile.

[‡]to whom correspondence should be addressed

open-source C++/Python library which extends the Boost Graph Library to be used as a complex network analysis workbench. The Boost Graph Library comprises an efficient, well-tested and standard-compliant set of graph algorithms and data structures. This allows our library to handle large networks and therefore makes it suitable for conducting analysis in a wide range of scenarios. Furthermore, Conan includes many complex network analysis tools in one simple, modular and unified framework. It spares the scientist the need to use multiple software tools to manage a single problem and helps them to develop more network algorithms and tools.

2 Capabilities

Many algorithms have been developed for the study of complex networks. The following list summarises those implemented by our library.

2.1 Network generation

Conan is able to generate networks based on models published by different authors [3, 6, 9, 17], and some canonical models, such as lattices and rings.

2.2 Network inference

Network inference or reverse engineering estimates the interaction networks that underlies observed phenomena. Several different methods for this purpose have been developed and their performance has been compared [2]. We have implemented some of these in Conan:

- Clustering coexpression: The covariance of every pair of elements (nodes) in the system is used for selecting the relevant relations.
- Maximum entropy (MaxEnt) [12]: A procedure to identify the pairwise interaction network that has the highest probability of giving rise to the observed data.
- ARACNe [13]: The mutual information of each pair of elements in the system is used to infer the network. This approach eliminates indirect interactions that are frequently inferred by co-expression methods.

2.3 Topological properties

Conan can calculate multiple global network properties, such as average shortest path (L), average clustering coefficient (CC), degree distribution, connectance (C), and many others (see Figure 1). The most remarkable are:

- Topological entropy [6]: This measure quantifies the uncertainty we would have guessing the position of a stochastic walker that has been visiting the network for an infinite amount of time. It has been correlated to the robustness of the network under random attack. (H in Figure 1)

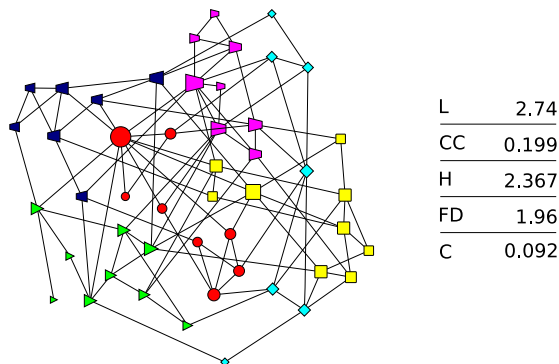


Figure 1: Genetic interaction network of the circadian clock of *Arabidopsis thaliana* inferred from microarray data using ARACNe, and some topological properties calculated with Conan. Nodes are module-wise coloured and proportional in size to their degree centrality. L, average shortest path; CC, average clustering coefficient; H, topological entropy; FD, fractal dimension; C, connectance.

- Fractal dimension [15]: This quantity, analogous to the euclidean fractal dimension, is an exponent which characterizes the self-similar structure of a network. It results from the power-law relation that appears under length-scale transformations of the network. (FD in Figure 1)

Conan can also calculate vertex and edge local properties, such as the vertex and edge centrality, and the vertex clustering coefficient, average shortest path, and eccentricity.

2.4 Subgraphs and communities

Conan allows the extraction of user-defined induced subgraphs from a network. It is also capable of splitting the network into communities using either the method designed by Newman [14] or by Guimera and Amaral [10].

2.5 Multiple format support

Conan is able to read and write in a variety of file formats, such as GraphML, DOT (<http://www.graphviz.org/>), GML [11], Pajek [4], and plain (ASCII) adjacency matrix files.

3 Interfaces

Conan is a native C++ library that provides an API for network generation, manipulation and analysis. The API was designed considering generic programming concepts, which avoids implementation-specific directives and encourages

the development of maximally reusable algorithms. But, in addition, our library can be utilized from the Python language (<http://www.python.org/>).

3.1 Python Bindings

This is the simplest way to access Conan, as it provides an intuitive object-oriented API, which lends itself to fast prototyping and simple inclusion within other software packages. The bindings' API was written paying close attention to Python's style guidelines.

4 Acknowledgements

This work was supported by Fundación Ciencia para la Vida; and Programa de Financiamiento Basal PFB16. We are grateful to Hector Urbina for contributing with code snippets.

References

- [1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, Jan 2002.
- [2] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. di Bernardo. How to infer gene networks from expression profiles. *Molecular Systems Biology*, 3, 2007. ISSN 1744-4292.
- [3] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. ISSN 00368075.
- [4] V. Batagelj and A. Mrvar. Pajek-program for large network analysis. *Connections*, 21(2):47–57, 1998.
- [5] I. Borodina and J. Nielsen. From genomes to in silico cells via metabolic networks. *Current Opinion in Biotechnology*, 16(3):350 – 355, 2005. ISSN 0958-1669.
- [6] L. Demetrius and T. Manke. Robustness and network evolution – an entropic principle. *Physica A: Statistical Mechanics and its Applications*, 346(3-4):682–696, 2005.
- [7] R. Demicheli and D. Coradini. Gene regulatory networks: a new conceptual framework to analyse breast cancer behaviour. *Annals of Oncology*, 2010.
- [8] N. C. Duarte, S. A. Becker, N. Jamshidi, I. Thiele, M. L. Mo, T. D. Vo, R. Srivas, and B. Ø. Palsson. Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proceedings of the National Academy of Sciences*, 104:1777–1782, 2007.

- [9] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.
- [10] R. Guimerà and L. A.N. Amaral. Cartography of complex networks: modules and universal roles. *Journal of Statistical Mechanics*, 2001:1–13, 2005.
- [11] M. Himsolt. GML – Graph Modelling Language. <http://www.fim.uni-passau.de/en/fim/faculty/chairs/theoretische-informatik/projects.html>, 1997.
- [12] T. R. Lezon, J. R. Banavar, M. Cieplak, A. Maritan, and N. V. Fedoroff. Using the principle of entropy maximization to infer genetic interaction networks from gene expression patterns. *Proceedings of the National Academy of Sciences*, 103(50):19033–19038, Dec 2006.
- [13] A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Favera, and A. Califano. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics*, 7(Suppl 1):S7, 2006.
- [14] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577, 2006.
- [15] C. Song, S. Havlin, and H. A. Makse. Self-similarity of complex networks. *Nature*, 433(7024):392–395, 2005.
- [16] S. H. Strogatz. Exploring complex networks. *Nature*, 410, 2001.
- [17] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.